2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

AdaM: Adaptive-Maximum Imputation for Neighborhood-based Collaborative Filtering

Yongli Ren, Gang Li*, Jun Zhang and Wanlei Zhou
School of Information Technology, Deakin University
221 Burwood Highway, Vic 3125, Australia
Email: {yongli, gang.li, jun.zhang, wanlei}@deakin.edu.au

***Abstract*—In the context of collaborative filtering, the well-known *data sparsity* issue makes two like-minded users have little similarity, and consequently renders the k nearest neighbour rule inapplicable. In this paper, we address the data sparsity problem in the *neighbourhood-based* CF methods by proposing an *Adaptive-Maximum imputation* method (AdaM). The basic idea is to identify an imputation area that can maximize the imputation benefit for recommendation purposes, while minimizing the imputation error brought in. To achieve the maximum imputation benefit, the imputation area is determined from both the user and the item perspectives; to minimize the imputation error, there is at least one real rating preserved for each item in the identified imputation area. A theoretical analysis is provided to prove that the proposed imputation method outperforms the conventional neighbourhood-based CF methods through more accurate neighbour identification. Experiment results on benchmark datasets show that the proposed method significantly outperforms the other related state-of-the-art imputation-based methods in terms of accuracy.**

I. Introduction

Collaborative Filtering (CF) is one of the most popular recommendation techniques, as it is insensitive to the detailed features of both users and items. Based on the applied techniques, it is generally categorized into two groups: the *Neighbourhood-based* CF methods and the *Model-based* methods [1], [2]. Recent lessons learnt from the *Netflix* prize competition reveal that these two kinds of methods can discover very different levels of data patterns in the $user \times item$ matrix, consequently none of them can provide the best results all the time based on its own [3]. As users tend to give ratings on only a small fraction of available items, the rating matrix is consequently sparse, which is one of the major challenges in CF methods. In this paper, we focus on the data sparsity problem in the context of neighbourhood-based CF methods.

The *Neighbourhood-based* CF methods are generally based on the k-nearest neighbour (KNN) rule. Namely, the neighbours' ratings are treated as *samples* of the unknown ratings of the active user that need to be predicted. However, the data sparsity issue makes the KNN rule stop working in the context of CF, as two-like minded users may not show any similarity in the sparse dataset. This will lead to challenges to apply this kind of CF methods into industrial areas that involve sparse data. One promising method to deal with this challenge is data imputation, to which this work belongs. *Default Voting* is a straightforward imputation-based method [4] that assumes default values for those missing ratings. Recently, Xue et. al. proposed the use of certain machine learning methods to smooth all missing data in the $user \times item$ rating matrix [5]. Ma et. al. took the imputation confidence into consideration, and only filled in the missing data when it was confident to impute [6]. This idea works well as it prevents poor imputation. However, all these work treats all the missing data equally. In another recent attempt, Ren et. al. proposed the Auto-Adaptive Imputation (AutAI) method, which can identify the purest possible neighbourhood for the active user [7]. However, due to the nature of data imputation, the imputation benefit and error should be treated collectively at the same time, namely the imputation benefit should be maximized while the imputation error should be minimized. In the context of neighbourhood-based CF methods, the question becomes, how to maximize the imputation benefit for each rating prediction adaptively? how to select the imputation area to guarantee the KNN rule working in this field? These are the questions that we investigate in this study.

In this paper, we propose an *Adaptive-Maximum imputation* (*AdaM*) method to maximize the imputation benefit for each rating prediction adaptively for the neighbourhood-based CF methods. The basic idea is to identify the imputation area by determining the maximum possible item set and the maximum possible user set adaptively for each singular rating prediction. Through this, it is guaranteed that each determined item has at least one rating from the determined user set. Thus, when measuring the relationship between two users in the determined item space, there is at least one real rating to lead the analysis towards the correct direction so as to avoid misleading analysis caused by the imputation error brought in. A theoretical analysis is provided to prove that the proposed *AdaM* method can theoretically outperform the traditional neighbourhood-based CF methods in terms of accurate nearest neighbour identification. Furthermore, based on the *AdaM* method, we propose a CF algorithm by fusing the *AdaM* method from both the user and the item perspective (*AdaM-Fusion*). Inspired by Cover's [8] research on the nearest neighbour rule, which showed that the nearest neighbour contains at least 50% of the information in the infinite training set, it is rational that the proposed *AdaM* method will lead to better recommendation performance, which is also shown in the experiment results on two benchmark datasets under various data sparsity conditions.

* Corresponding Author.

ASONAM'13, August 25-29, 2013, Niagara, Ontario, CAN

The major contributions of this paper are as follows:

- An *Adaptive-Maximum imputation* method is proposed to maximize the imputation benefit for each rating prediction adaptively to address the data sparsity issue.
- Theoretical analysis is provided to prove the performance benefit of the proposed method.
- We further propose an *AdaM-Fusion* CF algorithm to integrate the *AdaM* method from both the user and the item perspective.
- A large set of experiments is conducted to evaluate the recommendation performance of the *AdaM-Fusion* method, e.g. on various benchmark datasets, on various similarity metrics, under various data sparsity conditions, and to compare it with 7 state-of-the-art CF algorithms, including other imputation-based methods and model-based CF methods.

The rest of the paper is organized as follows. In Section II, we define the *Adaptive-Maximum Imputation* (*AdaM*) method in the context of neighbourhood-based collaborative filtering, together with a theoretical analysis on the imputation benefit and error. Section III presents the experiment results, followed by the conclusion in Section IV.

II. The Adaptive-Maximum Imputation for Collaborative Filtering

Let us assume $\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$ be a set of m users, $\mathcal{T} = \{t_1, t_2, \cdots, t_n\}$ be the set of n items. For m users and n items, the $user \times item$ rating matrix $\mathcal{R}$ is represented as a $m \times n$ matrix. $\mathcal{R}$ can be decomposed into row vectors: $\mathcal{R} = [\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_m}]^T$ and $\mathbf{u_i} = [r_{i1}, r_{i2}, \cdots, r_{in}]$, where T denotes *transpose*. The row vector $\mathbf{u_i}$ corresponds to the user u_i's rating information, and r_{ij} denotes the rating that user u_i gave to item t_j. $\mathcal{N}_k(u_x)$ denotes the set of user u_x's k nearest neighbours. $T_{xy} = \{t_i \in T | r_{xi} \neq \emptyset, r_{yi} \neq \emptyset\}$ denotes the set of items co-rated by both u_x and u_y. u_a denotes the active user for whom the recommendation is processing. t_s denotes the active item on which the recommendation is processing. $\bar{u}_a$ and $\bar{t}_s$ denote the average rating of user u_a and the average rating on item t_s, respectively.

A. Formulation of Neighbourhood-based CF

We first formulate the *neighbourhood*-based CF using probability theory [9]. The *neighbourhood*-based CF provides recommendations by estimating how much the active user may like un-rated items, which is known as the *rating prediction* task. Given two variables, the user consuming history u and the available ratings r, the rating prediction task can be formulated as $\mu(u) = E(r|u)$, which is the expectation of dependent variable r given the independent variable u. For the recommendation purpose, it is interesting to estimate the value r_{as} on an unrated item t_s for a singular independent variable value u_a. The estimator for u_a is then:

$$\hat{\mu}(u_a) = E(r_{as}|u_a). \tag{1}$$

From the perspective of probability theory, the observation values sampled at u_a can be used to estimate $\hat{\mu}(u_a)$.

However, there are no observation values at u_a in the context of collaborative filtering. To tackle this problem, certain similar users of u_a are selected and used to estimate $\hat{\mu}(u_a)$. This is the assumption of collaborative filtering: *similar users may have similar ratings on items*. Specifically, to estimate $\hat{\mu}(u_a)$, we first select several *neighbours* of u_a, then treat the ratings of these neighbours as *samples* at u_a. Finally, we process the above estimation method as usual. This is the well-known k nearest neighbour estimator [2] (KNN):

$$\hat{\mu}(u_a) = E(r_{as}|u_a) = \frac{1}{k} \sum_{u_x \in \mathcal{N}_k(u_a)} r_{xs}, \tag{2}$$

where k is the number of selected neighbours, u_x is one of u_a's neighbours, $\mathcal{N}_k(u_a)$ is the set of k nearest neighbours, and r_{xs} is the rating of neighbour u_x on t_s. To further reduce the estimation bias, the KNN estimation usually applies the weighted average [10]:

$$\hat{\mu}(u_a) = E(r_{as}|u_a) = \frac{1}{\eta} \sum_{u_x \in \mathcal{N}_k(u_a)} w_{ax} r_{xs}, \tag{3}$$

where w_{ax} is the weight of neighbour u_x to u_a, and $\eta = \sum_{u_x \in \mathcal{N}_k(u_a)} w_{ax}$ is the normalizing factor. The weights reflect how close the neighbours are to u_a.

Recent research shows that the data sparsity issue brings much difficulty to the neighbourhood-based CF methods [11], [12]. The data sparsity issue can lead to the critical problem of unreliable similarity since the similarity between two users is actually calculated using a very small set of co-rated items. The user relationship measured by the unreliable similarity cannot capture the overall relationship between two users. Moreover, the similarities of an active user u_a to two users (u_x and u_y) are computed on two different sets of co-rated items, which results in incomparable similarities. Consequently, the performance of neighbourhood-based CF is seriously affected by inaccurate similarities due to the data sparsity issue.

B. An Adaptive-Maximum Imputation Method

In this section, we propose an adaptive-maximum imputation method (*AdaM*) to deal with the missing values in the $user \times item$ rating matrix for the neighbourhood-based CF methods. It is a fact that imputation-based methods will bring in both imputation benefit and error for recommendation purposes. To make a rating prediction on the active item t_s for the active user u_a, we argue that there is an upper boundary to achieve the largest imputation benefit while keeping the imputation error minimized.

The proposed *AdaM* method can identify the maximum possible item set that affects the estimation of the neighbourhood relationship for the active user. Specifically, to make rating prediction on item t_s for user u_a, the imputed area is determined by two factors, the maximum set of possible neighbours related to u_a and the maximum set of possible items related to the prediction on t_s for u_a. We define the maximum set of neighbour candidates as $\mathcal{U}_a$, and define the maximum set of related items as $\mathcal{T}_s$:

$$\mathcal{U}_a = \{u_{a'}|r_{a's} \neq \oslash\} \cup \{u_a\}, \quad (4)$$
$$\mathcal{T}_s = \{t_i|t_i \subset [S_a \cup S_{a'_1} \cup \cdots \cup S_{a'_i} \cup \cdots \cup S_{a'_l}]\}, \quad (5)$$

where $S_{a'_i}$ is the items rated by $u_{a'_i} \in \mathcal{U}_a$, and $l = |\mathcal{U}_a|$. For example, suppose that the rating histories for two users u_a and $u_{a'}$ are represented as:

$$\mathbf{u_a} = \begin{bmatrix} r_{a1}, & r_{a2}, & 0, & 0, & r_{a5}, & 0, & \cdots, & 0 \end{bmatrix}$$
$$\mathbf{u_{a'}} = \begin{bmatrix} r_{a'1}, & 0, & 0, & r_{a'4}, & r_{a'5}, & 0, & \cdots, & r_{a'n} \end{bmatrix},$$

where 0 denotes the missing values. Then $\mathcal{T}_s = [t_1, t_2, t_4, t_5, \cdots, t_n]$, since they are rated at least once by one user. Therefore, $\mathcal{T}_s$ is the union of $S_{a'}$ over all $u_{a'} \in \mathcal{U}_a$. It is clear that this maximizes the possible item set while guaranteeing that each item has at least one real rating. Furthermore, with respect to item t_s, we define the *max neighbourhood* for the active user u_a for the prediction on t_s as:

$$\mathcal{N}_{a,s} = \{r_{a'i}|u_{a'} \in \mathcal{U}_a, t_i \in \mathcal{T}_s\}, \quad (6)$$

where $r_{a'i}$ can either be an observing or missing rating. Since the $user \times item$ rating matrix is sparse, this selected max neighbourhood is also sparse. We define all the missing data in this max neighbourhood as the *key set* of missing data for the prediction $\hat{r}_{as}$. Every observing rating $r_{a'i}$ in $\mathcal{N}_{a,s}$, is very important in the prediction for $\hat{r}_{as}$, because both user $u_{a'}$ and item t_i have direct relationship to $\hat{r}_{as}$. Following this this, the missing data in $\mathcal{N}_{a,s}$ also possess equivalent information for the prediction of $\hat{r}_{as}$, since they forms the maximum possible item spaces to measure the relationship among the active user and neighbour candidates. As $\mathcal{N}_{a,s}$ is defined from the user's perspective, it is termed as the *user*-based adaptive-maximum imputation method (*user*-based *AdaM*). When $\mathcal{N}_{a,s}$ is identified, the following equation is applied to impute each individual missing value $r_{a'i} \in \mathcal{N}_{a,s}$:

$$\hat{r}_{a'i} = \bar{u}_{a'} + \frac{\sum_{u_x \in \mathcal{N}_k(u_{a'})} sim(u_{a'}, u_x) \times (r_{xi} - \bar{u}_x)}{\sum_{u_x \in \mathcal{N}_k(u_{a'})} sim(u_{a'}, u_x)}, \quad (7)$$

where $sim(u_{a'}, u_x)$ is the PCC [13] similarity between $u_{a'}$ and u_x. After this, we use both the imputed ratings and the observing ratings in $\mathcal{N}_{a,s}$ to estimate the relationship among the active user and the neighbour candidates. As clearly presented, the key set of missing data is identified based on each individual user and item so as to maximize the imputation benefit in the max neighbourhood. We provide a theoretical analysis for this in Section II-C.

On the other hand, *AdaM* can also work in an item-based manner. Specifically, the *max neighbourhood* is identified for the item perspective, and can be formally defined as $\mathcal{N}'_{a,s} = \{r_{a'i}|u_x \in \mathcal{U}'_a, t_{s'} \in \mathcal{T}'_s\}$, where $\mathcal{T}'_s = \{t_{s'}|r_{as'} \neq \oslash\} \cup \{t_s\}$, $\mathcal{U}'_a = \{u_x|u_x \subset [U_s \cup U_{s'_1} \cup \cdots \cup U_{s'_i} \cup \cdots \cup U_{s'_l}]\}$, $U_{s'_i}$ denotes the set of users who rated $t_{s'_i}$, and $l = |\mathcal{T}'_s|$. The missing data in $\mathcal{N}'_{a,s}$ forms the *key set* of missing data in this item-based manner. We call this the *item*-based adaptive-maximum imputation method (*item*-based *AdaM*).

The imputation of the identified key missing values in $\mathcal{N}_{a,s}$ can effectively contribute to the recommendation purposes. It is well-known, from Cover's [8] research on the nearest neighbour rule, that the nearest neighbour contains at least 50% of the information in the infinite training set. Therefore, the identified missing data in $\mathcal{N}_{a,s}$ are more informative than the data outside $\mathcal{N}_{a,s}$. So, it is reasonable to obtain much more benefit from the imputation of these missing data. At the same time, the introduced imputation errors is also fewer than that from other related techniques. A performance analysis about the imputation benefit is presented in the following section. Moreover, according to the assumption of *neighbourhood*-based CF algorithms that *if two users rated n items similarly, they tend to rate similarly on other items* [4], *AdaM* makes the similarity measurement between the active user and any other neighbour on the same $\mathcal{T}_s$ items, rather than on the co-rated items between two users. Formally, the similarity between two users u_a and u_x on imputed data can be measured by any similarity metric. For instance, its PCC-based version is formulated as:

$$sim'(u_a, u_x) = \frac{\sum_{t_i \in \mathcal{T}_s} (r_{ai} - \bar{u}_a)(r_{xi} - \bar{u}_x)}{\sqrt{\sum_{t_i \in \mathcal{T}_s} (r_{ai} - \bar{u}_a)^2 \sum_{t_i \in \mathcal{T}_s} (r_{xi} - \bar{u}_x)^2}}, \quad (8)$$

where $\bar{u}_x$ is the average rating of user u_x.

As the imputed missing data in $\mathcal{N}'_{a,s}$ also contributes in a similar way, we will not list them separately. Following Eq. 8, the similarity between two items in the item-based *AdaM*, is defined in a similar way, e.g. its PCC-based version is defined as:

$$sim'(t_s, t_i) = \frac{\sum_{u_{a'} \in \mathcal{U}'_a} (r_{a's} - \bar{t}_s)(r_{a'i} - \bar{t}_i)}{\sqrt{\sum_{u_{a'} \in \mathcal{U}'_a} (r_{a's} - \bar{t}_s)^2 \sum_{u_{a'} \in \mathcal{U}'_a} (r_{a'i} - \bar{t}_i)^2}}, \quad (9)$$

where $\bar{t}_s$ is the average rating of item t_s.

C. Performance Analysis

In this section, we provide a theoretical analysis to answer why and how the proposed *AdaM* method benefits the conventional neighbourhood-based CF methods. Particularly, as the quality of determined selected neighbours dominate the performance of KNN-based CF methods, we study the performance analysis from the perspective of accurate nearest neighbour identification.

Let us consider a general case between three users, u_x, u_y and u_z in $\mathcal{U}_a$, and they have the following rating histories:

$$\mathbf{u_x} = [r_{x1}, \hat{r}_{x2}, \cdots, \hat{r}_{xi}, \cdots, \hat{r}_{xl}]$$
$$\mathbf{u_y} = [\hat{r}_{y1}, r_{y2}, \cdots, r_{yi}, \cdots, r_{yl}]$$
$$\mathbf{u_z} = [\hat{r}_{z1}, r_{z2}, \cdots, \hat{r}_{zi}, \cdots, \hat{r}_{zl}]$$

where $\hat{r}_{xi}$ is the imputed rating on item t_i for u_x. The rating history for each user can be divided into two parts, the available part $\mathbf{u_x^a}$ and the missing part $\mathbf{u_x^m}$,respectively.

Therefore, $\mathbf{u_x}$ and $\mathbf{u_y}$ can be represented as:

$$\begin{aligned}\mathbf{u_x} &= [\overbrace{r_{x(1)}, \cdots, r_{x(p)}}^{|\mathbf{u_x^a}|}, \overbrace{\hat{r}_{x(p+1)}, \cdots, \hat{r}_{x(l)}}^{|\mathbf{u_x^m}|}] \\ \mathbf{u_y} &= [\overbrace{r_{y(1)}, \cdots, r_{y(q)}}^{|\mathbf{u_y^a}|}, \overbrace{\hat{r}_{y(q+1)}, \cdots, \hat{r}_{y(l)}}^{|\mathbf{u_y^m}|}].\end{aligned}$$

In this study, we apply the covariance to measure the similarity between two users in the item space. For the covariance value between two users, u_x and u_y, it is defined as:

$$\begin{aligned}\delta_{xy}^{adam} &= E[(u_x - E[u_x])(u_y - E[u_y])] \\ &= \frac{1}{n}\sum_{i=1}^{n}(r_{xi} - E[u_x])(r_{yi} - E[u_y]). \quad (10)\end{aligned}$$

Thus, the observation on each item can be considered as an estimation for the covariance. For u_x and u_y, the observed covariance estimation on item t_i can be formulated as:

$$\zeta_{t_i} = (r_{xi} - E[u_x])(r_{yi} - E[u_y]). \quad (11)$$

Therefore,

$$\delta_{xy}^{adam} = \frac{1}{n}\sum_{i=1}^{n}(r_{xi} - E[u_x])(r_{yi} - E[u_y]) = \frac{1}{n}\sum_{i=1}^{n}\zeta_{t_i}. \quad (12)$$

In addition, we assume that this observation on any item is independent and identically distributed (i.i.d), and we adopt the normal distribution for theoretical analysis. Then, the Probability Density Function (PDF) [9], [14] of ζ_{t_i} is:

$$p(\zeta_{t_i}) \sim \begin{cases}\mathcal{N}(\mu_1, \sigma_1^2), & \text{for } u_x \text{ and } u_y \\ \mathcal{N}(\mu_2, \sigma_2^2), & \text{for } u_x \text{ and } u_z.\end{cases} \quad (13)$$

Similarly, we assume the imputation error ε on any item for any user is also independent and identically distributed (i.i.d), and its PDF is:

$$p(\varepsilon) \sim \mathcal{N}(\mu_\varepsilon, \sigma_\varepsilon^2). \quad (14)$$

For each user, the observed difference on each item can be considered as an estimation for the deviation to his/her mean value. For user u_x, the observed difference on item t_i is denoted as $d_{t_i}^x$. Then, the deviation d_{t_i} on item t_i for u_x can be expressed by

$$d_{t_i}^x = \begin{cases} r_{xi} - E[u_x], & \text{if } r_{xi} \text{ is real,} \\ \hat{r}_{xi} - E[u_x] = r_{xi} + \varepsilon_i - E[u_x], & \text{if } r_{xi} \text{ is imputed,}\end{cases} \quad (15)$$

where $\varepsilon_i = r_{xi} - \hat{r}_{xi}$ denotes the imputation error on item t_i for user u_x, and $\hat{r}_{xi}$ denotes the imputed value.

For the expectation value, $E(u_x)$, for u_x, it can be practically estimated by using the average of all observations, so,

$$E(u_x) = \frac{1}{n}\sum_{i=1}^{n} r_{xi}, \quad (16)$$

where n is the number of items taking into account when measuring $E(u_x)$. In the user-based *AdaM* method, this set of items is $\mathcal{T}_s$, and $E(u_x)^{adam}$ denotes the observed expectation $E(u_x)$. Considering $\mathcal{T}_s$ consists of two subsets, $\mathbf{u_x^a}$ and $\mathbf{u_x^m}$, $E(u_x)^{adam}$ is affected by items coming from both of them, and can be represented as:

$$\frac{1}{p}\sum_{i=1}^{p} r_{xi} \text{ and } \frac{1}{q}\sum_{j=1}^{q} \hat{r}_{xj},$$

where $p = |\mathbf{u_x^a}|$, $q = |\mathbf{u_x^m}|$, and $\hat{r}_{xj} = r_{xj} + \varepsilon_j$ represents the estimation from imputed values as defined in Eq. 15. Due to the existence of the imputation error ε, the estimation coming from $\mathbf{u_x^m}$ has to take it into consideration. The *cumulative imputation error* for the estimation of $E(u_x)^{adam}$ over $\mathbf{u_x^m}$ is:

$$\begin{aligned}\varepsilon_{adam} &= \frac{1}{q}\sum_{j=1}^{q}\hat{r}_{xj} - \frac{1}{q}\sum_{j=1}^{q} r_{xj} \\ &= \frac{1}{q}\left(\sum_{j=1}^{q}(r_{xj} + \varepsilon_j) - \sum_{j=1}^{q} r_{xj}\right) \\ &= \frac{1}{q}\sum_{j=1}^{q}\varepsilon_j. \quad (17)\end{aligned}$$

Consequently, $E(u_x)^{adam}$ in *AdaM* is represented as the calculation based on real ratings plus the *cumulative imputation error* ε_{adam}:

$$\frac{1}{l}\sum_{i=1}^{l} r_{xi} \text{ and } \varepsilon_{adam} = \frac{1}{q}\sum_{j=1}^{q}\varepsilon_j,$$

where $l = |\mathcal{T}_s|$. Consequently, the measurement of $E(u_x)^{adam}$ over $\mathcal{T}_s$ is defined:

$$E(u_x)^{adam} = \frac{1}{l}\sum_{i=1}^{l} r_{xi} + \frac{1}{q}\sum_{j=1}^{q}\varepsilon_j = \bar{u}_x + \frac{1}{q}\sum_{j=1}^{q}\varepsilon_j, \quad (18)$$

where $l = |\mathcal{T}_s|$ and $\bar{u}_x = \frac{1}{l}\sum_{i=1}^{l} r_{xi}$.

Then, the deviation d_{t_i} on item t_i for u_x can be expressed by

$$d_{t_i}^x = \begin{cases} r_{xi} - \bar{u}_x - \frac{1}{q}\sum_{j=1}^{q}\varepsilon_j, & \text{if } r_{xi} \text{ is real rating,} \\ r_{xi} - \bar{u}_x - \frac{1}{q-1}\sum_{j=1}^{q-1}\varepsilon_j, & \text{if } r_{xi} \text{ is imputed.}\end{cases} \quad (19)$$

According to Eq. 14 and Eq. 17, ε_{adam} is also normal, and its PDF is:

$$p(\varepsilon_{adam}) \sim \mathcal{N}(\mu_\varepsilon, \frac{\sigma_\varepsilon^2}{q}), \quad (20)$$

where $q = |\mathbf{u_x^m}|$. Normally, due to $|\mathbf{u_x^m}| \gg 1$, $\frac{\sigma_\varepsilon^2}{q}$ is much smaller than σ_ε^2. For example, in *MovieLens* data set, $mean(|\mathbf{u_x^m}|) = 3000$, which means the order of magnitude of $\frac{\sigma_\varepsilon^2}{q}$ is around three orders of magnitude less than σ_ε^2. Namely, ε_{adam} actually varies little around its mean value, and its variance can be ignored comparing with its initial distribution ε. Then, we obtain

$$d_{t_i}^x = r_{xi} - \bar{u}_x - \mu_\varepsilon. \quad (21)$$

Similarly, the deviation d_{t_i} on item t_i for u_y is obtained as:

$$d_{t_i}^y = r_{yi} - \bar{u}_y - \mu_\varepsilon. \quad (22)$$

Thus, we obtain the covariance value between two users, u_x and u_y in $\mathcal{R}_{as}$ as:

$$\begin{aligned}\delta_{xy}^{adam} &= E[(u_x - E[u_x])(u_y - E[u_y])] \\ &= \frac{1}{l}\sum_{i=1}^{l}(r_{xi} - \bar{u}_x - \mu_\varepsilon)(r_{yi} - \bar{u}_y - \mu_\varepsilon) \\ &= \frac{1}{l}\sum_{i=1}^{l}\zeta_{t_i} + \mu_\varepsilon^2, \end{aligned} \tag{23}$$

where $l = |\mathcal{T}_s|$. According to Eq. 13, Eq. 14 and Eq. 18, we obtain that:

$$p(\delta_{xy}^{adam}) \sim \mathcal{N}(\mu_1 + \mu_\varepsilon^2, \frac{\sigma_1^2}{l}). \tag{24}$$

Similarly, the PDF of the distance between u_x and u_z in AdaM, δ_{xz}^{adam}, is:

$$p(\delta_{xz}^{adam}) \sim \mathcal{N}(\mu_2 + \mu_\varepsilon^2, \frac{\sigma_2^2}{l}). \tag{25}$$

We define the *distance divergence* ψ between the distance of two users u_y and u_z to user u_x as:

$$\psi = \delta_{xy} - \delta_{xz}, \tag{26}$$

which determines their order to u_x. It is well-known that the *confidence interval* (CI) can be used to indicate the reliability of an estimation [9], and has been widely used in the research of neighbourhood-based models [10]. Therefore, in this study, we apply CI to measure the reliability of the estimation of ψ. CI is a range of values that quantify the uncertainty of the estimation, and a narrow CI means high precision [9]. According to Eq. 24 and Eq. 25, the $100(1-\alpha)\%$ confidence interval for ψ in *AdaM* can be formulated as follows:

$$\begin{aligned}CI(\psi_{adam}) &= \left((\mu_1 + \mu_\varepsilon^2) - (\mu_2 + \mu_\varepsilon^2)\right) \\ &\quad \pm z_{\alpha/2}\sqrt{\frac{\frac{\sigma_1^2}{l}}{l} + \frac{\frac{\sigma_2^2}{l}}{l}} \\ &= (\mu_1 - \mu_2) \pm z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{l^2} + \frac{\sigma_2^2}{l^2}}, \end{aligned} \tag{27}$$

where $z_{\alpha/2}$ is a standard normal variate which is exceeded with a probability of $\alpha/2$. We define $\sigma_{\psi_{adam}}$ as the standard error of ψ_{adam}:

$$\sigma_{\psi_{adam}} = \sqrt{\frac{\sigma_1^2}{l^2} + \frac{\sigma_2^2}{l^2}}. \tag{28}$$

Please note the width of $CI(\psi_{adam})$ is proportional to $\sigma_{\psi_{adam}}$.

Now, let us consider conventional neighbourhood-based CF, the measurement of the distance between u_x and u_y, δ_{xy}^{knn}, is estimated over T_{xy}, which is:

$$\delta_{xy}^{knn} = E(d_t) = \frac{1}{p}\sum_{i=1}^{p} d_{t_i}, \tag{29}$$

where $p = |T_{xy}|$. According to Eq. 13 and Eq. 29, we have

$$p(\delta_{xy}^{knn}) \sim \mathcal{N}(\mu_1, \frac{\sigma_1^2}{p_1}), \tag{30}$$

where $p_1 = |T_{xy}|$. Similarly, δ_{xz}^{knn}, the distance between u_x and u_z in conventional neighbourhood-based CF, also follows the normal distribution, and its PDF is:

$$p(\delta_{xz}^{knn}) \sim \mathcal{N}(\mu_2, \frac{\sigma_2^2}{p_2}), \tag{31}$$

where $p_2 = |T_{xz}|$. Therefore, according to Eq. 30 and Eq. 31, the $100(1-\alpha)\%$ confidence interval for ψ_{knn}, ψ in neighbourhood based CF approaches, is:

$$\begin{aligned}CI(\psi_{knn}) &= (\mu_1 - \mu_2) \pm z_{\alpha/2}\sqrt{\frac{\frac{\sigma_1^2}{p_1}}{p_1} + \frac{\frac{\sigma_2^2}{p_2}}{p_2}} \\ &= (\mu_1 - \mu_2) \pm z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{p_1^2} + \frac{\sigma_2^2}{p_2^2}}, \end{aligned} \tag{32}$$

where $z_{\alpha/2}$ is a standard normal variate which is exceeded with a probability of $\alpha/2$. $\sigma_{\psi_{knn}}$ is the standard error of the estimated ψ_{knn}:

$$\sigma_{\psi_{knn}} = \sqrt{\frac{\sigma_1^2}{p_1^2} + \frac{\sigma_2^2}{p_2^2}}, \tag{33}$$

where $p_1 = |T_{xy}|$ and $p_2 = |T_{xz}|$. And the width of $CI(\psi_{knn})$ is proportional to $\sigma_{\psi_{knn}}$.

According to Eq. 5, it is clear that $l = |\mathcal{T}_s| \geq |T_{xy}| = p_1$ and $l = |\mathcal{T}_s| \geq |T_{xz}| = p_2$. For example, in the benchmark dataset *MovieLens*, $mean(|\mathcal{T}_s|) = 3000$ and $mean(|T_{xy}|) = 18.4$. Together with Eq. 28 and Eq. 33, we obtain:

$$\sigma_{\psi_{adam}} \leq \sigma_{\psi_{knn}} \tag{34}$$

with equality if and only if $|\mathcal{T}_s| = |T_{xy}|$ and $|\mathcal{T}_s| = |T_{xz}|$, which is not valid when facing the data sparsity issue in the field of collaborative filtering. Furthermore, according to Eq. 27 and Eq. 32, one can see that the $CI(\psi_{adam})$ is narrower than or equal to $CI(\psi_{knn})$, as $\sigma_{\psi_{adam}}$ is smaller than or equal to $\sigma_{\psi_{knn}}$. Based on the above theoretical analysis, we conclude that the proposed *AdaM* method can effectively improve the performance of the conventional neighbourhood-based CF methods through more accurate nearest neighbour selections by using a sparse rating matrix in a novel way.

D. Rating Prediction

Based on the *AdaM* method, we proposed the *AdaM-Fusion* CF algorithm to predict ratings from both the user and the item perspective. In this way, the user activity and the item popularity can be fully integrated to provide better recommendations.

On one side, we adopt the user-based *AdaM* method to impute the missing values from the user perspective, in which the maximum set of users' (including possible neighbours) activities will be exploited as described in Eq. 5. Then, the prediction for r_{as} in this user-based manner is calculated as:

$$\hat{r}_{as}^{u} = \bar{u}_a + \frac{\sum_{u_x \in \mathcal{N}_k(u_a)} sim'(u_a, u_x) \times (r_{xs} - \bar{u}_x)}{\sum_{u_x \in \mathcal{N}_k(u_a)} sim'(u_a, u_x)}, \tag{35}$$

where $\bar{u}_a$ is the average rating of u_a, and $sim'(u_a, u_x)$ is defined in Eq. 8. On the other side, to take item popularity into consideration, we do an item-based *AdaM* imputation as described in Section II-B, then the prediction for r_{as} in this item-based manner is calculated as:

$$\hat{r}^i_{as} = \bar{t}_s + \frac{\sum_{t_i \in \mathcal{N}_k(t_s)} sim'(t_s, t_i) \times (r_{ai} - \bar{t}_i)}{\sum_{t_i \in \mathcal{N}_k(t_s)} sim'(t_s, t_i)}, \quad (36)$$

where $\bar{t}_s$ is the average rating of t_s, and $sim'(t_s, t_i)$ is defined in Eq. 9.

The final prediction for r_{as} in the proposed *AdaM-Fusion* algorithm is calculated as a linear integration:

$$\hat{r}_{as} = \lambda \hat{r}^u_{as} + (1 - \lambda)\hat{r}^i_{as}, \quad (37)$$

where λ is a predefined parameter to determine to what extend to integrate the predictions from the user and the item perspective, and its value can be determined by doing cross-validation. A similar fusion strategy is performed in [6], [7], [15], [16]. Specifically, the final prediction will be totally determined by performing the user-based *AdaM* imputation when $\lambda = 1$, while it will be generated by taking the item-based *AdaM* imputation when $\lambda = 0$. The effect of λ on the performance of AdapM-Fusion will be discussed in Section III-C.

III. EXPERIMENT

The data set we experiment with is the popular benchmark datasets *MovieLens* and *Netflix*. The *MovieLens* includes around 1 million ratings collected from $6,040$ users on $3,900$ movies. Following literature [17], the *Netflix* dataset is a subset extracted from the Netflix Prize dataset. in which each user rated at least 20 movies and each movie was rated by $20 - 250$ users. Specifically, we also apply the *All-But-One* configuration, in which we randomly select one single rating for each user in the datasets, then try to predict its value when observing all the other ratings the user has given.

Furthermore, of particular interest in Collaborative Filtering research is the relationship between data sparsity and generated recommendations. To evaluate the performance thoroughly, we extract a subset of 2000 users from *MovieLens* who rated at least 30 movies, and further we set up several different experimental configurations. Specifically, we split the selected subset into two sets, the Training set and the Test set. The size of the Training set varies from the first 500, 1000 and 1500 users, which are denoted as M_{500}, M_{1000} and M_{1500} respectively. The remaining 500 users are treated as the Test set. For each of the active user within the Test set, we alter the number of rated items provided from 10, 20 to 30, which are represented as $Given_{10}$, $Given_{20}$ and $Given_{30}$, respectively. This protocol is widely used in Collaborative Filtering research [5], [6], [15].

For consistency with other literatures [5], [6], [15], we apply the *Mean Absolute Error* (MAE) as the measurement metric, which is defined as:

$$MAE = \frac{\sum_{(a,s) \in X} |r_{as} - \hat{r}_{as}|}{|X|}, \quad (38)$$

TABLE I: Comparison on different similarity measurement metrics on the $Given$ Data set. (A smaller value means better performance)

Training	Users	PCC	AdaM+PCC	COS	AdaM+COS
M_{500}	$Given_{10}$	0.7925	0.7440	0.7855	0.7640
	$Given_{20}$	0.7663	0.7331	0.7663	0.7475
	$Given_{30}$	0.7546	0.7287	0.7561	0.7390
M_{1000}	$Given_{10}$	0.7802	0.7369	0.7853	0.7605
	$Given_{20}$	0.7594	0.7266	0.7672	0.7442
	$Given_{30}$	0.7491	0.7223	0.7563	0.7361
M_{1500}	$Given_{10}$	0.7764	0.7352	0.7854	0.7598
	$Given_{20}$	0.7581	0.7240	0.7687	0.7434
	$Given_{30}$	0.7487	0.7193	0.7585	0.7355

TABLE II: Comparison on different similarity measurement metrics on the *All-But-One*. (A smaller value means better performance)

Training Users	PCC	AdaM+PCC	COS	AdaM+COS
MovieLens	0.7477	0.6529	0.7652	0.6640
Netflix	0.7339	0.6545	0.7559	0.6585

where r_{as} is the true rating given by user u_a on item t_s, $\hat{r}_{as}$ is the predicted rating, X denotes the test data set, and $|X|$ represents its size. A smaller MAE value means better performance.

A. Performance with Different Similarity Metrics

As the similarity measurement metric is a key factor in neighborhood-based CF methods, we examine the performance of the proposed *AdaM* method on two state-of-the-art similarity metrics in Collaborative Filtering [1], [2], [5], [6], [15], the PCC and the COS. To get a fair comparison, we implement all compared algorithms in this section in the user-based manner. Specifically, we compare the user-based *AdaM* with the user-based CF algorithm and the user-based CF algorithm, on PCC and COS, respectively. The number of neighbours is set to 30. We measure the performance on all the experiment configurations, including $Given$ and *All-But-One* data sets.

Table I shows the MAE results on the $Given$ datasets. It is observed that, *AdaM* achieves significant improvement on both PCC and COS similarity metrics over all data sparsity levels. This demonstrates that *AdaM* can identify the neighbouring relationship among users more accurately, and is robust the data sparsity issue. Specifically, on the $M_{500}Given_{10}$ dataset when measuring in PCC, *AdaM* achieves a MAE at 0.7440 outperforming the counterpart 0.7925 by 6.12%.

Moreover, the MAE results on the *All-But-One* dataset are shown in Table II. For PCC, the MAE is reduced from 0.7477 to 0.6529 by 12.68%, and for COS, it is reduced from 0.7652 to 0.6640 by 13.23%. The improvement in this *All-But-One* dataset is larger than that in the $Given$ datasets. The reason behind is that, there is no restriction to user's rating history, then the *AdaM* can work maximumly to enhance the information in the neighborhood of the active user; on the

TABLE III: Statistics of Paired-t-test on PCC and COS

	Paired-t statistics		
Methods	df	t	p-value
AdaM+PCC vs. PCC	10	6.5752	< 0.0001
AdaM+COS vs. COS	10	3.8285	0.0033

TABLE IV: Comparison with other methods on the $Given$ Datasets. (A smaller value means better performance)

Training users	Methods	$Given_{10}$	$Given_{20}$	$Given_{30}$
M_{500}	AdaM-Fusion	0.7390	0.7269	0.7221
	AutAI-Fusion	0.7579	0.7387	0.7298
	EMDP	0.7724	0.7573	0.7421
	SCBPCC	0.7837	0.7633	0.7550
	Default Voting	0.7762	0.7594	0.7513
	UPCC	0.7925	0.7663	0.7546
	IPCC	0.7927	0.7613	0.7493
	Slope One	0.7704	0.7455	0.7341
M_{1000}	AdaM-Fusion	0.7315	0.7203	0.7156
	AutAI-Fusion	0.7488	0.7298	0.7211
	EMDP	0.7602	0.7440	0.7301
	SCBPCC	0.7784	0.7602	0.7499
	Default Voting	0.7776	0.7614	0.7535
	UPCC	0.7802	0.7594	0.7491
	IPCC	0.7673	0.7402	0.7293
	Slope One	0.7601	0.7374	0.7263
M_{1500}	AdaM-Fusion	0.7294	0.7179	0.7130
	AutAI-Fusion	0.7445	0.7256	0.7172
	EMDP	0.7548	0.7383	0.7264
	SCBPCC	0.7792	0.7587	0.7506
	Default Voting	0.7812	0.7659	0.7580
	UPCC	0.7764	0.7581	0.7487
	IPCC	0.7572	0.7315	0.7214
	Slope One	0.7581	0.7355	0.7246

other side, the number of ratings for the active user is restricted to 10, 20 or 30, which is actually impressing the imputation ability of the *AdaM* method.

The two-tailed, paired t-test with a 95% confidence level has been applied to evaluate the performance of *AdaM* on both similarity metrics. The results show that the difference of performance with and without *AdaM* is statistically significant. The detailed paired-t statistics are shown in Table III.

B. Comparison with Other Methods

To examine the performance of the *AdaM-Fusion* algorithm, we compare it with other 7 state-of-the-art imputation-based algorithms, including the *default voting* (Default Voting) method [4], the EMDP method [6], the SCBPCC method [5], and AutAI-Fusion method [7], and two traditional collaborative filtering algorithms, the *user-based* CF (UPCC) and the *item-based* CF (IPCC), and one model-based algorithm, the Slope One algorithm [18]. The parameters in SCBPCC are set as $\lambda = 0.35$, and the cluster number $K = 20$. All the parameters in EMDP are set the same as [6], namely $\lambda = 0.7, \gamma = 30, \delta = 25, \eta = \theta = 0.4$. For AutAI-Fusion, λ is set to 0.4. Please note that EMDP and AutAI-Fusion are imputation-based algorithms by fusing the user-based CF algorithm and the item-based CF algorithm. The parameter λ in our algorithm is set to 0.4.

TABLE VI: Statistics of Paired-t-test with the State-of-the-art

	Paired-t statistics		
Methods	df	t	p-value
AdaM-Fusion vs. AutAI-Fusion	10	4.9119	0.0006
AdaM-Fusion vs. EMDP	10	5.0137	0.0005
AdaM-Fusion vs. SCBPCC	10	6.2631	0.0001
AdaM-Fusion vs. Default Voting	10	5.4473	0.0003
AdaM-Fusion vs. UPCC	10	8.0336	< 0.0001
AdaM-Fusion vs. IPCC	10	6.0810	0.0001
AdaM-Fusion vs. Slope One	10	4.1661	0.0019

Table IV shows the MAE results on the $Given$ datasets. We observe that *AdaM-Fusion* outperforms all the other 7 compared algorithms significantly on all datasets with various data sparsity conditions. For example, on the $M_{500}Given10$ data set, although all imputation-based methods (*AdaM-Fusion*, AutAI-Fusion, EMDP, SCBPCC and Default Voting) achieve a better performance than UPCC and IPCC, *AdaM-Fusion* achieves the largest improvement. This indicates that when taking both the item popularity and user activity into consideration, the performance of the *AdaM* method can be improved further when comparing the results in Table I. Thus, leads to the better performance that outperforms the other compared algorithms, including other imputation-based methods and model-based method. Moreover, the results on *MovieLens* and *Netflix* in the *All-But-One* setting are shown in Table V. It is also clearly observed that *AdaM-Fusion* achieves better performance than others on both dataset in terms of MAE. Specifically, on the *Netflix* dataset, *AdaM-Fusion* obtains the smallest MAE value 0.6512.

Table VI lists the paired-t test statistics (with a 95% confidence level) between *AdaM-Fusion* and these algorithms, and it is clear that the differences between the performance of *AdaM-Fusion* and that of other methods are statistically significant.

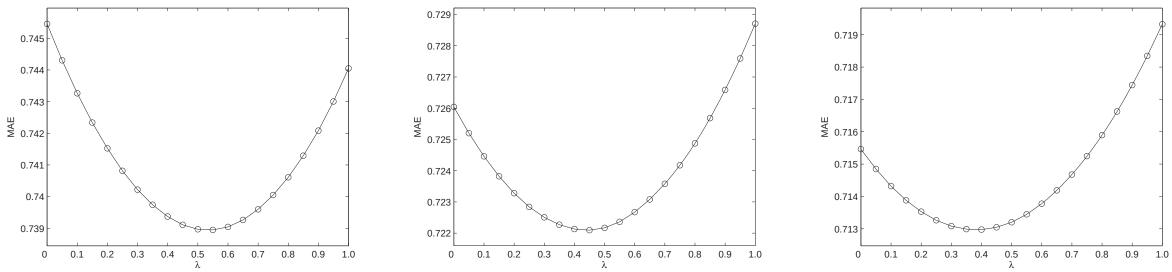
C. Impact of Parameter

In this study, to gain the imputation benefit as much as possible, we integrate the imputation effect from both the user and item perspective by introducing a parameter λ to balance the predictions from both of them. We simultaneously take the user activity and item popularity into consideration to determine the imputation area for a particular rating prediction. Here, We conducted several experiments to explore the effect of λ on the performance of the proposed *AdaM-Fusion* algorithm. Specifically, we keep $k = 30$ and vary the λ value from 0 to 1 with an increasing step of 0.05.

As defined in Eq. 37, the prediction totally depends on the item-based *AdaM* imputation when $\lambda = 0$; it will depends on the user-based imputation when $\lambda = 1$. The results on the $Given$ datasets are shown in Fig. 1. It is observed that better performance are obtained when taking both of user and item information into account, significantly outperforming the results from the results from either one of them. For example, on data sets $M_{500}Given_{10}$ and $M_{500}Given30$ as shown in Fig. 1a and Fig. 1b, the performance of *AdaM-Fusion* with $\lambda = 0.4$ is better than its performance with $\lambda = 0$ or 1. The

TABLE V: MAE comparison with other methods on the *All-But-One* Datasets. (A smaller value means better performance)

Training Users	AdaM-Fusion	AutAI-Fusion	EMDP	SCBPCC	Default Voting	UPCC	IPCC	Slope One
MovieLens	0.6528	0.6864	0.7244	0.7613	0.7954	0.7477	0.7056	0.7164
Netflix	0.6512	0.6830	0.7282	0.7590	0.7403	0.7339	0.7068	0.7347



(a) M_{500} $Given_{10}$ (b) M_{500} $Given_{30}$ (c) M_{1500} $Given_{30}$

Fig. 1: Impact of Parameter λ

reason is that when the user's rating profile is limited by the experiment setting, the item-based *AdaM* will perform better than the user-based one, and will complement the prediction from the user perspective. Yet, on all configurations, it is clear that better accuracy can be obtained by combining imputations from users and items. However, to achieve a fair comparison, we used a fix $\lambda = 0.4$ in *AdaM-Fusion* in Section III-B.

IV. Conclusion

In this paper, we address the data sparsity issue in the context of neighborhood-based CF methods by proposing the *AdaM* method to maximize the imputation benefit while minimizing the imputation error. Specifically, we treat each rating given by a user on an item as a potential sample for prediction on this item for other users, then the *AdaM* method is proposed to optimize the relationship between the active user and its potential neighbours by identifying the largest imputation area, which is defined in a way to guarantee that at least one real rating is observed on each selected item in the determined imputation area. A theoretical analysis is provided to prove the proposed method guarantee to achieve the maximum imputation benefit so as to identify the neighbourhood relationship more accurately than other the conventional neighbourhood-based CF methods, including other imputation-based methods. Furthermore, the *AdaM-Fusion* algorithm is proposed to take both the user activity and item popularity into account to complement the prediction from each of these two perspectives. The experiment results show that the proposed *AdaM-Fusion* algorithm can achieve better predictions than the state-of-the-art related algorithms.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE TKDE*, vol. 17, no. 6, pp. 734–749, 2005.

[2] C. Desrosiers and G. Karypis, "Chapter 4 A comprehensive Survey of Neighborhood-based Recommendation Methods," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, ch. 4, pp. 107–144.

[3] R. M. Bell and Y. Koren, "Lessons from the Netflix Prize Challenge," *SIGKDD Explorations*, vol. 9, no. 2, pp. 75–79, 2007.

[4] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, 1998, pp. 43–52.

[5] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *SIGIR 2005*. New York, USA: ACM Press, 2005, pp. 114–121.

[6] H. Ma, I. King, and M. R. Lyu, "Effective Missing Data Prediction for Collaborative Filtering," in *SIGIR 2007*, pp. 39–46.

[7] Y. Ren, G. Li, J. Zhang, and W. Zhou, "The efficient imputation method for neighborhood-based collaborative filtering," in *CIKM '12*, 2012, pp. 684–693.

[8] T. Cover, "Estimation by the Nearest Neighbor Rule," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 50–55, 1968.

[9] G. K. B. Richard A. Johnsom, *Statistics: Principles and Methods*, 6th ed. John Wiley and Sons, 2009.

[10] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, p. 175, Aug. 1992.

[11] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, 2009.

[12] Y. Ren, G. Li, and W. Zhou, "Learning Rating Patterns for Top-N Recommendations," *The 2012 IEEE/ACM International Conference on Social Networks Analysis and Mining (ASONAM)*, 2012, pp. 472–479.

[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens : An Open Architecture for Collaborative Filtering of Netnews," in *ACM Conference on Computer Supported Cooperative Work*, 1994, pp. 175–186.

[14] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network Traffic Classification Using Correlation information," *IEEE Transaction on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 104–117, 2013.

[15] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *SIGIR 2006*. New York, USA: ACM Press, 2006, pp. 501–208.

[16] H. Vu, G. Li, N. Sukhorukova, and G. Beliakov, "K-Complex Detection Using a Hybrid-Synergic Machine Learning Method," *IEEE Transactions on Systems, Man, and Cybernetics-PartC: Applications and Reviews*, vol. 42, no. 6, pp. 1478–1490, 2012.

[17] X. Ning and G. Karypis, "SLIM : Sparse Linear Methods for Top-N Recommender Systems," in *ICDM 2011*, 2011, pp. 497 – 506.

[18] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *In SIAM Data Mining (SDM'05)*, vol. 05, 2005.